

# Alcance en Grafos

## 1. Descripción de la actividad 1

Se desea crear una aplicación, para buscar un camino entre dos puntos en un mapa. Nuestro mapa urbano está compuesto por las intersecciones de la ciudad y las calles que los unen. Una calle puede ser doble vía o de un sólo sentido. Una consulta de una ruta consiste en un par  $\langle I, F \rangle$ , donde  $I$  es el punto de origen de la ciudad y  $F$  es el punto de llegada. Se quiere que la aplicación muestre la ruta a seguir, como una secuencia desde  $I$  hasta  $F$ , con la forma  $I X_0 X_1 \dots X_n F$  donde  $X_i$  es una intersección del mapa en la que incide la calle que va desde  $X_{i-1}$  a  $X_i$ . Se asume que el punto de origen y de llegada son diferentes. Debe implementar ésta aplicación en el lenguaje de programación Java, no se requiere que la ruta sea óptima, es decir, se quiere obtener cualquier ruta, sin importar su longitud, para esto se requerirá que su programa esté basado en el *Algoritmo de Roy-Warshall*.

Debe crear un archivo llamado Ruta.java con la implementación de la aplicación.

## 2. Entrada de datos

El programa se debe poder ejecutar desde la consola con el siguiente comando:

```
>java Ruta <instancia> <origen> <destino>
```

Donde instancia es el nombre del archivo que contiene un grafo que representa al mapa, *origen* es un número entero que corresponde al sitio de partida y *destino* es un número entero que representa el lugar de llegada. Todos los datos de entrada del archivo son números enteros. El formato del archivo que contiene los datos de un grafo es el siguiente:

NumeroDeVertice

NumeroDeLados

idVerticeInicial1 idVerticeFinal1

:

idVerticeInicialNumeroDeLado idVerticeFinalNumeroDeLado

## 3. Salida de los Datos

Si existe una ruta entre dos puntos, se debe imprimir la ruta como unas secuencias de intersecciones separadas por espacio  $I X_0 X_1 \dots X_n F$ . En caso de no haber una ruta, se muestra por la salida estándar el mensaje *No hay una ruta entre los dos puntos*.

## 4. Ejemplo

Dado los siguientes datos en el archivo de entrada facil.txt:

```
6
8
0 5
5 4
4 5
3 4
3 2
0 2
2 1
1 0
```

Se obtiene el siguiente resultado al correr la aplicación:

```
> java Ruta facil.txt 3 5
> 3 2 1 0 5

> java Ruta facil.txt 1 3
> No hay ruta entre los dos puntos
```

## 5. Pruebas

Adicionalmente se hace entrega del paquete JUnit3.8.1 (el archivo junit-3.8.1.jar) para que usted experimente y pueda realizar pruebas a su software de forma rápida. En el archivo TestRuta.java se encuentran escritas 30 casos de prueba en el formato del framework JUnit3.8.1, que permite que usted pueda ejecutar todas las pruebas y ver los resultados de forma automática. Usted debe asegurarse antes de entregar su laboratorio, que su programa pasa estas 30 pruebas.

Para compilar el archivo TestRuta.java escriba el siguiente comando

```
> javac -cp <ruta de junit-3.8.1.jar>:. TestRuta.java
```

Para desplegar la interfaz gráfica de JUnit escriba el siguiente comando

```
> java -cp <ruta de junit-3.8.1.jar>:. junit.swingui.TestRunner TestRuta &
```

Mantenga la ventana emergente abierta durante el desarrollo y ejecute las pruebas las veces que desee haciendo click en el botón *Run*. Usted puede ver cuales fueron las pruebas que paso el software y cuales no, en la pestaña *Test Hierarchy*.